
BACHELORARBEIT

Herr
Tobias Schaarschmidt

**Entwurf und Entwicklung einer
App für den mobilen Zugriff
auf ein cloudbasiertes
Bildungsmanagementsystem**

2016

BACHELORARBEIT

Entwurf und Entwicklung einer App für den mobilen Zugriff auf ein cloudbasiertes Bildungsmanagementsystem

Autor:
Herr Tobias Schaarschmidt

Studiengang:
Mobile Media

Seminargruppe:
MD13w1-B

Erstprüfer:
Prof. Dr.-Ing. Frank Zimmer

Zweitprüfer:
Dipl.-Inf. Henrik Würzberger

Einreichung:
Mittweida, 05.10.2016

BACHELOR THESIS

Design and development of an app for the mobile access to a cloudbased education management system

author:

Mr. Tobias Schaarschmidt

course of studies:

Mobile Media

seminar group:

MD13w1-B

first examiner:

Prof. Dr.-Ing. Frank Zimmer

second examiner:

Dipl.-Inf. Henrik Würzberger

submission:

Mittweida, 05.10.2016

Bibliografische Angaben

Schaarschmidt, Tobias

Entwurf und Entwicklung einer App für den mobilen Zugriff auf ein cloudbasiertes Bildungsmanagementsystem

Design and development of an app for the mobile access to a cloudbased education management system

38 Seiten, Hochschule Mittweida, University of Applied Sciences,
Fakultät Medien, Bachelorarbeit, 2016

Abstract

Das Ziel dieser vorliegenden Bachelorarbeit ist ein funktionstüchtiger Prototyp in Form einer mobilen Anwendung. Die Applikation greift auf ein Bildungsmanagementsystem zu. Wie die Software realisiert wurde und welche Technologien für Entwicklung verwendet wurden, wird in dieser Arbeit dargestellt.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	V
Abkürzungsverzeichnis	VII
Abbildungsverzeichnis.....	VIII
Tabellenverzeichnis.....	IX
1 Einleitung.....	1
1.1 Motivation.....	1
1.2 Zieldefinition / Ziele	1
1.3 Aufbau der Arbeit	2
2 Grundlagen.....	3
2.1 Bildungsmanagement	3
2.1.1 Bildungsmanagementsystem	4
2.2 Anwendungssoftware für Mobilgeräte	5
2.3 Cloud-Computing	8
2.3.1 Vorteile von Cloud-Computing.....	9
2.3.2 Nachteile von Cloud-Computing.....	10
2.3.3 Deutschland-Cloud.....	11
2.4 Weitere Begriffe	12
2.4.1 Werkzeuge.....	12
3 Praktische Umsetzung.....	14
3.1 Technologieentscheidung	14
3.1.1 Mobiles Betriebssystem	14
3.1.2 Programmiersprache.....	16
3.1.3 Serialisierung der Daten.....	20
3.2 Anforderungen	23
3.3 Realisierung des Prototyps	26
3.3.1 Kommunikationsweg	26
3.3.2 Methoden des Webserver	26
3.3.3 JSON Objekte	28
3.3.4 Oberfläche der mobilen Anwendung	31

4	Zusammenfassung und Weiterentwicklung.....	37
4.1	Zusammenfassung.....	37
4.2	Weiterentwicklung	37
	Glossar.....	X
	Literaturverzeichnis	XI
	Eigenständigkeitserklärung	XIV

Abkürzungsverzeichnis

API	Application Programming Interface
BMS	Bildungsmanagementsystem
JSON	JavaScript Object Orientation
XML	Extensible Markup Language
WWDC	Worldwide Developer Conference

Abbildungsverzeichnis

Abbildung 1: Bildungsmanagement-Prozesse	3
Abbildung 2: Nokia Communicator	5
Abbildung 3: erstes iPhone.....	6
Abbildung 4: Anzahl der Apps im Apple App Store	6
Abbildung 5: Anzahl der Apps im Google Play Store	7
Abbildung 6: Symboldarstellung Internet-Wolke	8
Abbildung 7: Cloud Abstraktionsschichten.....	8
Abbildung 8: Marktanteile mobiler Betriebssysteme	14
Abbildung 9: Fragmentierung der Android-Versionen	15
Abbildung 10: Fragmentierung der iOS-Versionen	16
Abbildung 11: Objective-C Quellcode Beispiel.....	18
Abbildung 12: Swift Quellcode Beispiel	18
Abbildung 13: Beispiel XML Format	20
Abbildung 14: Beispiel JSON Format	21
Abbildung 15: Kommunikationsweg.....	26
Abbildung 16: Überblick aller Views	31
Abbildung 17: Ansicht der Anmeldung.....	32
Abbildung 18: Fehlermeldung Anmeldung I.....	32
Abbildung 19: Fehlermeldung Anmeldung II.....	33
Abbildung 20: Ansicht der Benutzerdaten.....	33
Abbildung 21: Ansicht der Registrierungen.....	34
Abbildung 22: Ansicht der Registrierungen im Simulator	35
Abbildung 23: Ansicht des Gesamtangebots	35
Abbildung 24: Ansicht der Suchergebnisse im Simulator.....	36

Tabellenverzeichnis

Tabelle 1: Vergleich der Verfügbarkeit.....	11
Tabelle 2: Erläuterung Code-Beispiele	18
Tabelle 3: Rangliste Programmiersprachen.....	19
Tabelle 4: Vergleich Mobilfunkstandards	22
Tabelle 5: Anforderungen an Webserver	23
Tabelle 6: Anforderungen an Oberfläche des Prototyps	24
Tabelle 7: Anforderungen an Funktionen des Prototyps	25
Tabelle 8: Funktionen des Webserver	27
Tabelle 9: Erläuterung der JSON Objekte der Anfragen	28
Tabelle 10: Erläuterung der JSON Objekte der Antworten.....	30

1 Einleitung

1.1 Motivation

Der technische Fortschritt schreitet immer schneller voran. Moderne Technologien und die Digitalisierung eröffnen viele neue und nützliche Möglichkeiten, den Arbeitsalltag zu modernisieren. Eine besondere Rolle nehmen Behörden ein. Der Verwaltungsaufwand einiger Ämter ist enorm. Mithilfe fortschrittlicher Technik sind Behörden bestrebt, den Aufwand der Verwaltung zu reduzieren.

Aus diesem Grund haben die Polizeibehörden verschiedener Bundesländer in Zusammenarbeit mit der T-Systems Multimedia Solutions GmbH (T-Systems MMS) ein Bildungsmanagementsystem entworfen. Für Polizeibeamte gibt es eine Vielzahl an Weiterbildungen, welche zum Teil verpflichtend und regelmäßig durchzuführen sind. Mit dem Bildungsmanagementsystem werden die Aus- und Weiterbildungen verwaltet.

Die Leistungsfähigkeit mobiler Endgeräte ist in den letzten Jahren beachtlich gewachsen. Durch die Erfindung des Smartphones wurde das Verhalten der Handynutzung maßgeblich beeinflusst. Die Möglichkeit mit den mobilen Endgeräten ortsunabhängig auf das Internet zuzugreifen bietet große Anzahl neuer Perspektiven. Der Ausbau des mobilen Internets für zunehmend schnellere Übertragungsraten begünstigt die Nutzung des Smartphones unterwegs.

All diese Entwicklungen sind ideale Voraussetzungen, um eine praktische Umsetzung für den mobilen Zugriff auf ein Bildungsmanagementsystem zu realisieren.

1.2 Zieldefinition / Ziele

Ziel dieser Bachelorarbeit ist eine prototypische Anwendung, mit welcher berechnigte Personen in der Lage sein sollen, auf Teile des Bildungsmanagementsystems mit mobilen Endgeräten zuzugreifen.

Weiterhin soll geprüft werden, inwiefern sich eine Lösung, abseits von Behörden, für kleine und mittelständige Unternehmen eignet. Dabei ist zu beachten, dass Unternehmen dieser Art selten eigene Rechenzentren besitzen. Es soll erörtert werden, inwiefern das Bildungsmanagementsystem nicht nur für den lokalen Einsatz in der firmeneigenen Server-Infrastruktur, sondern für den Einsatz in der Cloud sinnvoll ist.

1.3 Aufbau der Arbeit

Kapitel 2 behandelt die Grundlagen für diese Arbeit. In Kapitel 2 wird zuerst der Begriff Bildungsmanagement erklärt. Dabei wird auf die geschichtliche Entwicklung eingegangen und erläutert, wie es sich im Laufe der Zeit gewandelt hat. Es folgt die Revolutionierung der mobilen Endgeräte und wie sie das Verhalten geändert haben. Zum Schluss des Kapitels wird das Thema Cloud-Computing erklärt und welche weiteren Begriffe für das Verständnis der Arbeit nötig sind. Verwendete Software und Werkzeuge werden kurz vorgestellt.

In Kapitel 3 wird die praktische Umsetzung beschrieben. Zunächst wird die Wahl der Technologie für die Realisierung des Prototyps betrachtet. Durch die Betrachtung der Vor- und Nachteile verschiedener mobiler Betriebssysteme, wird eine Entscheidung getroffen. Dazu gehört auch die Wahl der Programmiersprache. Es folgen die Anforderungen, welche die Anwendung erfüllen muss. Nach den Vorbetrachtungen wird die praktische Umsetzung der Software behandelt.

Die Schlussbetrachtung und Ideen zur Weiterentwicklung werden in Kapitel 4 behandelt.

2 Grundlagen

2.1 Bildungsmanagement

Bildungsmanagement beschreibt die Verwaltung und Planung von Bildungsangeboten. Es findet in den Bereichen Anwendung, wo Lehrinhalte bereitgestellt werden. Das kann eine Privatschule sein, in welcher neue Ausbildungen angeboten werden sollen. Oder ein Unternehmen, welches Bedarf sieht, die Mitarbeiter in neue Software einzuarbeiten. Um Bildungsinhalte zu organisieren, werden verschiedene Prozesse durchlaufen.



Abbildung 1: Bildungsmanagement-Prozesse¹

Am Anfang steht ein Bildungsbedarf, welcher durch verschiedene Faktoren entstehen kann. Das kann die Strategie einer Firma sein, die einen neuen Markt erschließen will und deshalb ausgebildetes Personal benötigt. Der Planungs-Prozess beginnt mit einer Bedarfsanalyse. Es wird der Qualifizierungs- und Personalentwicklungsbedarf ermittelt. Daraus werden Konzepte für die Gestaltung von Bildungsmaßnahmen entwickelt. Das umfasst die Zielsetzung der Bildungsmaßnahmen und die Konzeptionierung der Lehrinhalte. Die Vermarktung für die Zielgruppe ist dann von Bedeutung, wenn es sich bei den Unternehmen um Schulen handelt, welche mit den Bildungsmanagement-Prozessen ihre Kurse füllen wollen. Ist die Vermarktung abgeschlossen, beginnt die Durchführung der Bildungsmaßnahmen. Die Evaluierung beendet den Bildungsma-

¹ <http://www.diebildungsmanager.at/blog/bildungsmanagement/> , Abbildung 1

nagement-Prozess. Es wird geprüft, ob Lehrinhalte erfolgreich vermittelt wurden. Während der Durchführung erzielte Erkenntnisse, fließen in die weitere Planung wieder ein.

Dieser gesamte Prozess lässt sich an dem Beispiel einer Hochschule darstellen. Durch Veränderung der Wirtschaft werden Fachkräfte in neuen Bereichen benötigt. Eine Hochschule ist für die Ausbildung geeignet und sollte demzufolge Studiengänge anbieten, welche Studenten für geeignete Positionen ausbildet. Dazu muss eine Marktbeobachtung stattfinden, um so zeitnah wie möglich, Studiengänge zu entwickeln. Öffnet sich der Markt für neue Arbeitsplätze, müssen dazu Konzepte entwickelt werden. Bei der Konzeptionierung sollte darauf geachtet werden, dass die in dem Beruf benötigten Kenntnisse in die Planung mit aufgenommen wird. Bei der Organisation des Lehrprogramms sind geeignete Dozenten zu engagieren um die Lehrinhalte zu vermitteln. Mit dem Abschluss des Modulhandbuchs, ist der Konzept-Prozess beendet. Anschließend muss die Hochschule den Studiengang bekannt machen. Sind genug Studenten eingetragen, wird der Studiengang durchgeführt. Die Durchführung der Lehrinhalte richtet sich nach dem festgelegten Modulhandbuch. Nach dem Abschluss einzelner Module, sollten diese evaluiert werden. Dadurch werden Kenntnisse gewonnen, welche in die weitere Planung wieder einfließen können.

2.1.1 Bildungsmanagementsystem

Um die Bildungsmanagement-Prozesse abzubilden hat die T-Systems MMS in Zusammenarbeit mit den Polizeiämtern verschiedener Bundesländer ein Bildungsmanagementsystem (BMS) entwickelt.

Das System umfasst eine Veranstaltungsverwaltung, Fortbildungskatalog, Buchungsprozess, E-Learning, Fortbildungs-Historie und Berichte sowie Statistiken.

Der hohe Verwaltungsaufwand wurde mit dem BMS gesenkt. Die Software übernimmt Aufgaben, welche vorher manuell erledigt werden mussten. Mit der Veranstaltungsverwaltung werden Kurse erstellt mit allen nötigen Informationen. Der Buchungsprozess ermöglicht den Anwendern, sich für die Kurse zu registrieren.

2.2 Anwendungssoftware für Mobilgeräte

Mobile Anwendungen werden umgangssprachlich auch *App* genannt. Vor der Ära der Smartphones existierten Handys, auf welchen Software installiert werden konnte. Meist waren diese Geräte jedoch nicht für den produktiven Einsatz geeignet.



Abbildung 2: Nokia Communicator²

Es gab zum Beispiel von Nokia mobile Geräte, welche viele Funktionen mitbrachten, wie Kalender, Adressbuch, Notizen. Auch konnte mit diesen, eigene Software installiert werden. Die Leistung ist jedoch mit heutigen Smartphones nicht vergleichbar. Die Apps, wie es sie heute gibt, begannen 2008. Am 10.07.2008 eröffnete Apple den *App Store*³. Den Grundstein für den Bedarf an Apps legte Apple am 09.01.2007, als sie das iPhone der Öffentlichkeit vorstellten.⁴

² <http://www.apfelwiki.de/briefkasten/Main/nokia9300i.jpg> , Abbildung 1

³ Vgl.: <https://techcrunch.com/2008/07/10/app-store-launches-upgrade-itunes-now/> , Zeile 1

⁴ Vgl.: <http://www.apple.com/pr/library/2007/01/09Apple-Reinvents-the-Phone-with-iPhone.html> , Zeile 1



Abbildung 3: erstes iPhone⁵

Die neuartige Bedienung des Smartphones mit dem Finger und das relativ große Display im Vergleich zu den damaligen Geräten boten viel Platz um mit dem Gerät auch Webseiten aufzurufen.

Mit der Veröffentlichung des App Stores, waren Entwickler in der Lage, eigene Anwendungen über den App Store zu veröffentlichen.

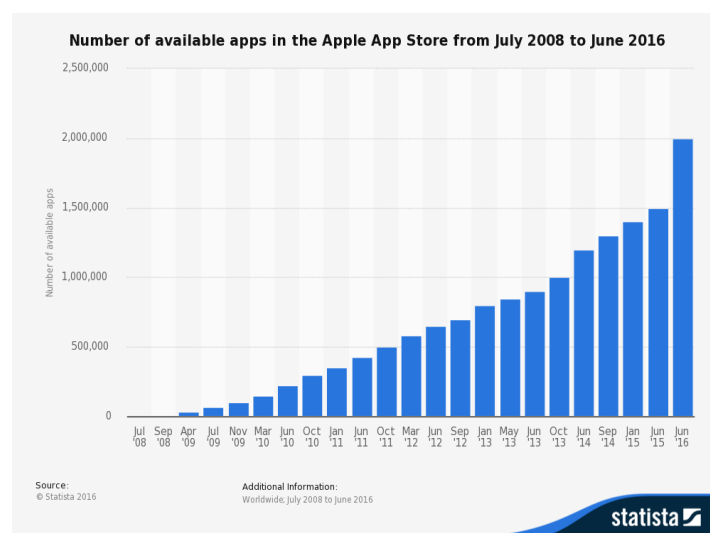


Abbildung 4: Anzahl der Apps im Apple App Store⁶

⁵ <https://www.iphoneshopusa.com/c/12-category/iphone-2g.jpg> , Abbildung 1

⁶ <https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/> , Abbildung 1

In Abbildung 4 wird die Anzahl der Anwendungen im App Store von Apple dargestellt. Von Juli 2008 bis Oktober 2011 wurden 500.000 Apps im Store veröffentlicht. Zwei Jahre später lag die Anzahl bei einer Million. Die Menge wurde bis Juni 2016 noch einmal verdoppelt. Durch diese Statistik wird erkennbar, dass es eine enorme Nachfrage für mobile Anwendungen gibt. Durch die leistungsfähigen Geräte und die qualitativ besserwerdenden Displays, ist es möglich auch produktiv damit zu arbeiten.

Mobile Applikationen gibt es nicht nur iOS. Am 23.09.2008 wurde das erste Smartphone mit *Android* vorgestellt.⁷ Knapp einen Monat später wurde auch für das Android-Betriebssystem ein Online-Marktplatz mit der Bezeichnung „Android Market“ den Anwendern zur Verfügung gestellt.⁸ Der Name für den Marktplatz war für Google nicht mehr aussagekräftig, deshalb wurde der „Android Market“ in „Google Play“ am 06.03.2012 umbenannt. Da in dem Marktplatz vorher auch E-Books, Musik und Videos zum Verkauf eingeführt wurden.⁹

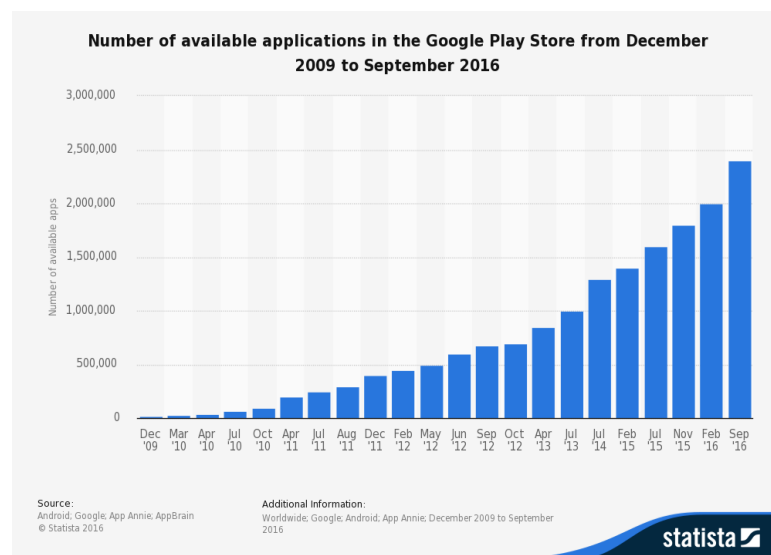


Abbildung 5: Anzahl der Apps im Google Play Store¹⁰

Der Vergleich der beiden Grafiken Abbildung 4 und Abbildung 5 verdeutlicht das exponentielle Wachstum. Bei Google Play gab es im September annähernd 2,5 Millionen Apps zum Download.

⁷ Vgl.: <http://www.giga.de/smartphones/htc-dream/news/>

htc-dream-erstes-android-smartphone-wurde-vor-5-jahren-vorgestellt/, Zeile 1 – 2

⁸ Vgl.: http://www.pcworld.com/article/152613/google_android_ships.html, Zeile 1

⁹ Vgl.: <https://techcrunch.com/2012/03/06/goodbye-android-market-hello-google-play/>, Zeile 1

¹⁰ <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>, Abbildung 1

In Kapitel „3.1.1 Mobiles Betriebssystem“ werden für die Technologieentscheidung beide Betriebssysteme näher untersucht.

2.3 Cloud-Computing

Als Cloud-Computing wird die Bereitstellung von IT-Ressourcen über das Internet bezeichnet. Häufig wird Cloud-Computing nur als Cloud bezeichnet. Der Begriff „Cloud“ lässt sich zurückführen auf die technischen Organigramme, in welchen das „Internet“ häufig als Wolke dargestellt wird.

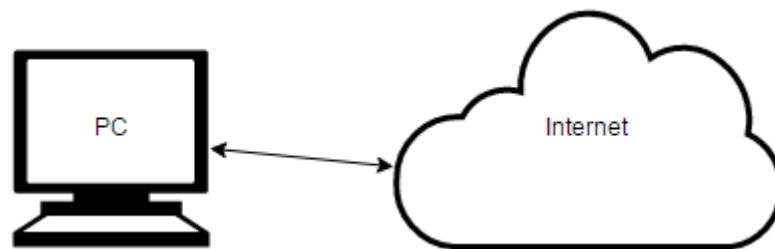


Abbildung 6: Symboldarstellung Internet-Wolke¹¹

Abbildung 6 stellt einen PC dar, welcher mit dem Internet verbunden ist. Aus diesen Organigrammen hat sich später der Begriff Cloud gebildet, da die IT-Infrastruktur über das Internet erreichbar ist.



Abbildung 7: Cloud Abstraktionsschichten¹²

¹¹ <https://www.draw.io/> , eigene Darstellung

¹² <https://blog.botfrei.de/2012/12/cloud-computing-oder-was-ist-eigentlich-die-cloud/> , Abbildung 2

Es gibt diverse Cloud-Anbieter, welche Dienste auf verschiedenen Ebenen anbieten. Diese Ebenen lassen sich unterteilen in „Software as a Service“ (SaaS), „Platform as a Service“ (PaaS) und „Infrastructure as a Service“ (IaaS).

SaaS ist die oberste Ebene bei Cloud-Diensten. Dabei werden Softwareanwendungen in der Cloud für Endnutzer bereitgestellt. In den meisten Fällen reicht ein aktueller Internetbrowser aus, um die Software auszuführen. Ein Beispiel dafür ist die Internetseite „<https://www.draw.io>“. Mit dieser Online-Anwendung wurde Abbildung 6 erstellt.

Unter SaaS ist die Ebene PaaS angesiedelt. Auf dieser Ebene stehen den Kunden Entwicklungswerkzeuge zur Verfügung. Somit können die Entwickler direkt in den Cloud-Umgebungen Software erstellen.

Die unterste Ebene ist IaaS. IaaS erfordert den größten Administrationsaufwand. Der Kunde hat die Möglichkeit Rechenleistung, Speicherplatz und andere Dinge manuell zu verwalten. Die Preise für diese Ebene richten sich nach verwendeter Rechenkapazität sowie Menge des Speicherplatzes.

2.3.1 Vorteile von Cloud-Computing

Die Vorteile von Cloud-Computing lassen sich aus verschiedenen Blickwinkeln betrachten.

Vorteile für Endanwender

Für die Software in der Cloud, wird in den meisten Fällen nur ein aktueller Internetbrowser benötigt. Das Arbeiten mit verschiedenen Betriebssystemen stellt kein Problem dar. Für die Nutzung der Programme muss nichts auf dem eigenen System installiert werden. Dadurch wird kein Speicherplatz auf dem Gerät des Anwenders in Anspruch genommen. Wird die Software auf verschiedenen Geräten ausgeführt, kann die Arbeit mit dem letzten Speicherstand fortgeführt werden, da es in der Cloud gespeichert wird. Daraus ergibt sich ein weiterer Vorteil, denn durch die Speicherung der Daten auf mehreren Datenträgern, erhöht sich die Ausfallsicherheit. Außerdem gibt es die Möglichkeit, dass zum Beispiel Dokumente in der Cloud gleichzeitig von verschiedenen Personen bearbeitet werden können.

Vorteile für Entwickler

Viele Vorteile für Entwickler decken sich mit denen der Endanwender. Denn durch die Bereitstellung verschiedener Entwicklungswerkzeuge in der Cloud, sind die Program-

mierer in der Lage, direkt im Browser zu entwickeln. Da die Webserver die meiste Rechenleistung übernehmen, sind auch schwächere Geräte zur Softwareentwicklung geeignet. Wird die Software für Endanwender in der Cloud bereitgestellt, lässt sich die Produktpiraterie senken. Die Software kann nicht mehr ohne Weiteres von Anwendern kopiert werden, da der Quellcode auf dem Server ausgeführt wird und nicht auf den Geräten der Endanwender.

Vorteile für Unternehmen

Für Unternehmen liegt der größte Vorteil darin, dass kein eigenes Rechenzentrum gebaut werden muss. Denn gerade für kleine und mittelständige Unternehmen, wäre dies in den meisten Fällen nicht realisierbar. Sehr große Unternehmen haben durchaus die Mittel, ein eigenes Rechenzentrum zu bauen. Für Firmen gibt es dafür die „Infrastructure as a Service“ Ebene. Die Auslagerung zu einem Cloud-Anbieter hat den Vorteil, dass die benötigten Ressourcen relativ unkompliziert skaliert werden können.

Durch die Redundanz der Speichersysteme, wird die Sicherheit der Daten erhöht. In den Rechenzentren existieren zudem hohe Sicherheitsstandards, welche den unbefugten Zugriff verhindern. Außerdem sind Sicherheitssysteme für Brände und ähnliches vorhanden. Große Rechenzentren sind auch durch mehrere unabhängige Stromleitungen vor einem Stromausfall abgesichert.

Für kleine Unternehmen sind solche Infrastrukturen in ihrer Gesamtheit nicht erreichbar, da die finanziellen Mittel nicht zur Verfügung stehen. Deshalb kann die Auslagerung bestimmter Teile der eigenen IT in die Cloud eine sinnvolle Alternative sein.

2.3.2 Nachteile von Cloud-Computing

Der größte Nachteil ist die Auslagerung der Daten in ein Netzwerk, über welche ein Unternehmen, aber auch Endanwender, keine Kontrolle haben. Zwar müssen sich die Cloud-Anbieter an Gesetze und Datenschutzbestimmungen halten. Sitzt der Anbieter aber im Ausland, gelten dort andere Gesetze und Bestimmungen als in Deutschland.

Ein weiterer Nachteil sind Ausfälle jeglicher Art. Die Schuld muss dabei nicht zwingend bei dem Rechenzentrum liegen. Nutzt ein Anwender eine Software in der Cloud, muss er demzufolge mit dem Internet verbunden sein. An dieser Stelle reicht ein Ausfall des Internetproviders des Anwenders aus, um die Arbeit zu unterbrechen. Auch durch fehlerhafte Konfigurationen der Server können wichtige Daten unwiderruflich gelöscht werden. Im April 2011 kam es zu solch einem Fehler des Cloud-Computing-Anbieters

Amazon Web Services. Während eines Netzwerkupgrades führte ein Routingfehler zu Ausfällen und Datenverlust.¹³

Verfügbarkeit in Prozent	Ausfallzeit pro Jahr in Minuten
99	5.256,0
99,9	525,6
99,95	262,8
99,99	52,56

Tabelle 1: Vergleich der Verfügbarkeit

Tabelle 1 zeigt die Unterschiede in der Betriebszeit bzw. der Verfügbarkeit der Server. Die Cloud-Anbieter garantieren verschiedene Verfügbarkeiten. Wie wichtig dabei die Stellen nach dem Komma sind, verdeutlicht die Tabelle. Eine garantierte Verfügbarkeit von 99 Prozent bedeutet, dass pro Jahr die Webserver über 87 Stunden nicht erreichbar sein können. Selbst bei 99,9 Prozent liegt die Nichtverfügbarkeit bei acht Stunden und 45 Minuten.

2.3.3 Deutschland-Cloud

Die KPMG AG hat in Zusammenarbeit mit dem Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. (Bitkom) 2015 eine Studie zum Thema Cloud-Computing in Deutschland durchgeführt. Dabei wurde unter anderem festgestellt, dass 60 Prozent der Unternehmen Angst haben, dass es zu einem unberechtigten Zugriff auf sensible Unternehmensdaten kommen könnte. Außerdem erwarten 83 Prozent der Kunden, dass ihr Cloud-Anbieter seine Rechenzentren ausschließlich in Deutschland betreibt.¹⁴

Um den Unternehmen die Angst zu nehmen und mehr Vertrauen zu gewinnen, entstand eine Partnerschaft mit Microsoft und der T-Systems. Microsoft übernimmt die Rolle des Cloud-Anbieters. Die Daten werden ausschließlich in Rechenzentren in Deutschland gespeichert. Die T-Systems fungiert als Datentreuhänder. Das heißt, als unabhängiges Unternehmen, kontrolliert die T-Systems den Zugang zu den Kundenda-

¹³ Vgl.: <http://www.golem.de/1104/83140.html> , Zeile 1 – 4

¹⁴ Vgl.: <https://www.bitkom.org/Publikationen/2015/Studien/Cloud-Monitor-2015/Cloud-Monitor-2015-KPMG-Bitkom-Research.pdf> , Seite 5, Abbildung 1

ten. Microsoft hat keinerlei Zugang zu den Kundendaten ohne Zustimmung des Datentreuhänders oder des Kunden. Kundendaten werden nur mit Zustimmung des Kunden herausgegeben oder wenn das deutsche Recht es verlangt.¹⁵

Microsoft setzt sich zum Ziel, eine transparente Cloud zu schaffen, um Unternehmen von ihrem Vorteil zu überzeugen. Die Nachfrage nach einem sicheren und vor allem transparenten Cloud-Anbieter ist groß. Microsoft versucht zusammen mit der T-Systems diesen Markt zu erreichen.

Um die Sicherheit der Daten zu gewährleisten, werden die Daten in zwei Rechenzentren bei Magdeburg und in Frankfurt am Main gespeichert und verarbeitet. Der Datenaustausch findet über ein vom öffentlichen Internet getrenntem Netzwerk statt. Um das Risiko eines Datenverlustes zu minimieren, findet ein kontinuierlicher Datenabgleich zwischen beiden Rechenzentren statt. Beide Rechenzentren unterliegen hohen Sicherheitsmaßnahmen mit Überwachung und Sicherheitsdienst rund um die Uhr.¹⁶

2.4 Weitere Begriffe

2.4.1 Werkzeuge

Für die Entwicklung des Prototyps werden verschiedene Softwarewerkzeuge eingesetzt.

Webserver

Die Programmierung der Software auf dem Webserver findet mit Visual Studio statt. Visual Studio ist eine Entwicklungsumgebung, welche von Microsoft angeboten wird. Die Erweiterung des Bildungsmanagementsystems wird mit der Programmiersprache C# geschrieben.

¹⁵ Vgl.: <https://www.microsoft-politik.de/de-de/politik/themen/cloud-deutschland.aspx>

¹⁶ Vgl.: <https://windowsunited.de/2016/09/21/microsoft-azure-die-deutschland-cloud-ist-da/> (07.08.2016)

Mobile Anwendung

Die Entwicklung des Prototyps erfolgt mit Xcode. Xcode ist eine von Apple bereitgestellte Entwicklungsumgebung. Mit dieser Software können unter anderem Anwendungen für das mobile Betriebssystem iOS erstellt werden. Die verfügbaren Sprachen sind hierbei Objective-C und Swift. Beide Sprachen und welche für die Umsetzung gewählt wurde, wird in Kapitel „3.1.2 Programmiersprache“ erläutert.

3 Praktische Umsetzung

Dieses Kapitel beschreibt die Realisierung der mobilen Anwendung. Es wird zunächst abgewogen, für welches mobile Betriebssystem die App umgesetzt wird. Weiterhin wird erörtert, welche Programmiersprache dafür geeignet ist. Es wird erläutert, wie die Kommunikation zwischen Anwender, mobilem Endgerät und Webserver stattfindet.

Anschließend werden die Funktionen des Webserver und der Applikation behandelt.

3.1 Technologieentscheidung

3.1.1 Mobiles Betriebssystem

Vor dem Beginn der Realisierung der App, wird zunächst geprüft, welches mobile Betriebssystem geeignet ist. Dazu werden vorerst die Marktanteile näher betrachtet.

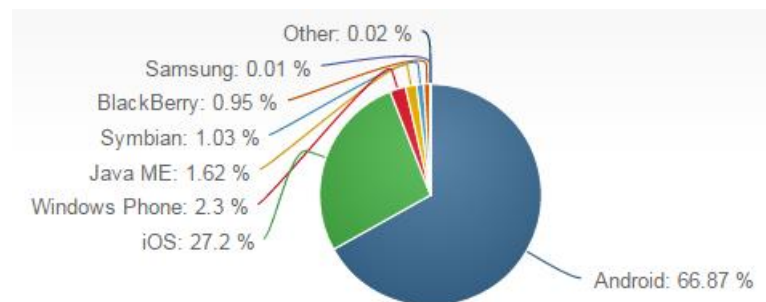


Abbildung 8: Marktanteile mobiler Betriebssysteme¹⁷

Das Diagramm in der Grafik stellt die Marktanteile mobiler Betriebssysteme dar. Es wird deutlich, dass Android von Google und iOS von Apple die zwei meist genutzten Betriebssysteme für Smartphones und Tablets sind. Zusammen dominieren beide den Markt mit ca. 94 Prozent Marktanteil. Aufgrund der niedrigen Verbreitung der anderen Systeme, werden diese für die weitere Betrachtung ausgeschlossen.

Für die Entscheidung zwischen Android und iOS werden die einzelnen Versionen im Folgenden untersucht. Beide Unternehmen bieten dazu eigene Statistiken an.

¹⁷ <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1> , Abbildung 1

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.4%
4.1.x	Jelly Bean	16	5.6%
4.2.x		17	7.7%
4.3		18	2.3%
4.4	KitKat	19	27.7%
5.0	Lollipop	21	13.1%
5.1		22	21.9%
6.0	Marshmallow	23	18.7%

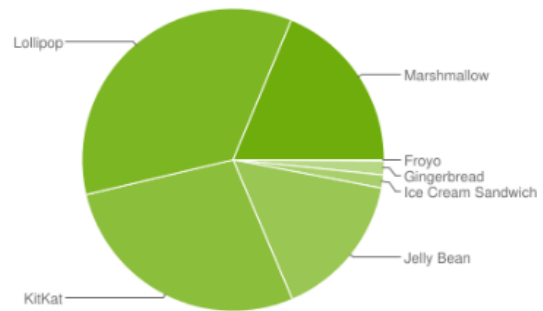


Abbildung 9: Fragmentierung der Android-Versionen¹⁸

Die Grafik zeigt die Verteilung der einzelnen Android-Versionen. Die Daten stammen direkt von Google. Sie wurden vom 30.08.2016 bis 05.09.2016 erfasst. Es werden nur Android-Versionen aufgelistet, welche einen Anteil von mehr als 0,1 Prozent besitzen. Interessant ist dabei, dass obwohl Android 7.0 am 22.08.2016 veröffentlicht wurde, es nicht in der Statistik auftaucht¹⁹. Das heißt, dass die aktuellste Version von Android auf nicht einmal 0,1 Prozent der Geräte installiert ist.

Auf über einem Viertel der Endgeräte ist Android 4.4 installiert. Die letzte Aktualisierung erhielt Android 4.4.4 am 19.06.2014²⁰. Zum Zeitpunkt dieser Arbeit ist diese Version mehr als zwei Jahre alt. Werden alle Veröffentlichungen bis Android 5.1 betrachtet, so ergibt sich eine Summe von 81,3 Prozent. Android 5.1.1 wurde am 21.04.2015 freigegeben²¹. Es wird deutlich, dass auf über 80 Prozent aller in der Statistik erfassten Android-Geräte, eine Software installiert ist, die mindestens eineinhalb Jahre alt ist.

¹⁸ <https://developer.android.com/about/dashboards/index.html> , Abbildung 1

¹⁹ <https://android.googleblog.com/2016/08/android-7.0-nougat-more-powerful-os-made.html> , Zeile 1

²⁰ <http://www.xda-developers.com/heres-everything-thats-changed-in-android-4-4-4-ktu84p/> , Zeile 1

²¹ <http://phandroid.com/2015/04/21/android-5-1-1-lmy47v-aosp-changelog-nexus-player-factory-images/> , Zeile 1

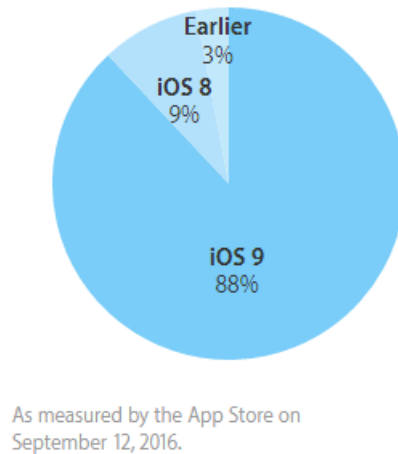


Abbildung 10: Fragmentierung der iOS-Versionen²²

Apple stellt auch eine Auswertung zur Verfügung. Die Daten wurden an einem Tag ermittelt. Die in der Grafik dargestellten Zahlen, wurden am 12.09.2016 gemessen. Es wurden alle Geräte protokolliert, welche an dem Tag den App Store aufgerufen haben. Zum Zeitpunkt der Messung ist iOS 9 die aktuellste Version und auf 88 Prozent der Geräte installiert.

Die Statistiken offenbaren, dass auf Smartphones und Tablet mit iOS die Unterstützung seitens des Herstellers besser ist, als auf den Geräten, auf denen Android installiert ist.

3.1.2 Programmiersprache

Bei der nativen Softwareentwicklung für iOS stellt Apple zwei Programmiersprachen zur Verfügung. Die erste Möglichkeit ist Objective-C. Objective-C wurde in den 1980er Jahren entwickelt und anschließend von Apple weiterentwickelt. Zur Worldwide Developer Conference (WWDC) 2006 wurde Objective-C 2.0 vorgestellt und im Oktober 2007 veröffentlicht.

Die zweite Möglichkeit ist die relativ neue Programmiersprache Swift. Swift wurde auf der WWDC 2014 vorgestellt. Ein Jahr später, auf der WWDC 2015, stellte Apple Swift 2.0 vor. Dabei wurden unter anderem sprachliche Erweiterungen präsentiert. Auch wurde die Lizenz auf eine Open Source Lizenz geändert.

²² <https://developer.apple.com/support/app-store/> , Abbildung 1

Objective-C war viele Jahre die Standardsprache für die Entwicklung von iOS Apps. Allerdings ist die Sprache schon mehrere Jahrzehnte alt und ist aus diesem Grund nicht mehr modern. Der große Vorteil von Objective-C ist die Verfügbarkeit von Drittanbieter *Frameworks*.

Swift wurde während der Entwicklung von verschiedenen Programmiersprachen beeinflusst. Ziel der Entwickler ist die Syntax schlank zu gestalten, so dass die Lesbarkeit hoch ist. Gleichzeitig sollte sich Swift auch in der Geschwindigkeit bei der Ausführung des Quellcodes gegenüber Objective-C verbessern. Apple hat dazu mit dem „depth-first search“ (DFS) Algorithmus Messungen durchgeführt. Es wurde ermittelt, dass Swift dabei bis zu 2,6-mal schneller als Objective-C ist²³. Am 03.12.2015 hat Apple bekannt gegeben, dass Swift ab sofort unter einer Open Source Lizenz steht²⁴. Dazu wurde Swift auf *GitHub* veröffentlicht, wodurch die komplette Entwicklung für jeden einsehbar ist. Somit sind alle Änderungen an der Sprache nachvollziehbar und transparent. Es besteht die Möglichkeit, die Programmiersprache zu beeinflussen, da das Swift-Entwicklerteam auf Verbesserungsvorschläge eingeht. Zwar sind viele Frameworks von Drittanbietern in Objective-C programmiert, allerdings besteht die Möglichkeit, diese auch in Swift-Projekten zu nutzen. Dadurch müssen diese Frameworks nicht zwingend für Swift entwickelt werden.

Jedoch gibt es nicht nur Vorteile. Ein großes Problem ist die Inkompatibilität zwischen den einzelnen Swift Versionen. Mit jeder neuen Version gibt es so starke Veränderungen an der jungen Sprache, dass der Quellcode einer mit Swift 1.0 entwickelten Anwendung nicht auf Version 2.0 ausführbar ist. Auch von Swift 2.3 auf 3.0 musste der Quellcode aktualisiert werden. Am 13.09.2016 wurde Swift 3.0 veröffentlicht²⁵. In der aktuellen Version wurden über 100 Neuerungen eingepflegt. Ein Ziel von Swift 3.0 ist die Festigung möglichst vieler Strukturen, so dass in den nachfolgenden Aktualisierungen der Quellcode kompatibel zu den Neuerungen ist. Die Inkompatibilität bedeutet aber nicht, dass veröffentlichte Apps nicht mehr mit neueren iOS Versionen funktionieren. Eine mit Swift 1.0 programmierte App, welche im App Store verfügbar ist, lässt sich auch auf aktuellen Geräten ausführen. Um die Entwickler bei der Aktualisierung ihrer Projekte zu unterstützen, bietet Apple eine automatische Konvertierung des Quellcodes auf die aktuelle Version an. Handelt es sich um ein großes und komplexes Projekt, ist bei der Konvertierung mit Fehlern zu rechnen, welche manuell beseitigt werden müssen.

Zur Verdeutlichung der Unterschiede in der Syntax zwischen Objective-C und Swift folgt ein kurzes Beispiel.

²³ Vgl.: <http://www.apple.com/de/swift/>

²⁴ <https://developer.apple.com/swift/blog/?id=34> , Zeile 1

²⁵ <https://swift.org/blog/swift-3-0-released/> , Zeile 1

```
NSString *benutzer = @"Max Mustermann";
int alter = 30;
NSString *s = [NSString stringWithFormat:@"Der Benutzer %@ ist %d Jahre alt.", benutzer, alter];
NSLog(@"%@", s);
```

Abbildung 11: Objective-C Quellcode Beispiel

Abbildung 11 ist ein Code-Beispiel in Objective-C.

```
let benutzer = "Max Mustermann"
let alter = 30
let s = "Der Benutzer \(benutzer) ist \(alter) Jahre alt."
print(s)
```

Abbildung 12: Swift Quellcode Beispiel

Abbildung 12 zeigt ein Code-Beispiel in Swift.

Zeile	Beschreibung
1	<ul style="list-style-type: none"> Anlegen einer Variable „benutzer“ vom Typ NSString/string mit dem Text „Max Mustermann“
2	<ul style="list-style-type: none"> Anlegen einer Variable „alter“ vom Typ int mit der Zahl „30“
3	<ul style="list-style-type: none"> Anlegen einer Variable „s“ vom Typ NSString/string Variable „s“ wird mit einem Text gefüllt, welcher die Variablen „benutzer“ und „alter“ angibt
4	<ul style="list-style-type: none"> Die Variable „s“ wird ausgegeben mit folgendem Inhalt: „Der Benutzer Max Mustermann ist 30 Jahre alt.“

Tabelle 2: Erläuterung Code-Beispiele

Tabelle 2 beschreibt die Vorgänge in den Zeilen der Code-Beispiele. In Zeile 1 und 3 steht „NSString“ für Objective-C und „string“ für Swift. Bei diesem einfachen Beispiel wird der Unterschied zwischen Swift und Objective-C deutlich. Während bei Objective-C Variablen einen Typ zugewiesen bekommen müssen, ist das in Swift nicht unbedingt nötig. Auch bemerkenswert ist die einfache Gestaltung zusammengesetzter Zeichenketten, was in Zeile 3 ausgeführt wird.

In Objective-C werden Platzhalter in Zeichenketten mit einem Prozentzeichen angegeben, gefolgt von einem „@“ für Zeichenketten oder „d“ für Ganzzahlen, je nach Typ der Variable. Am Ende der Zeichenkette müssen die Variablen kommasetrennt aufgezählt werden. Es ist zu beachten, dass die Variablen in der gleichen Reihenfolge aufgelistet werden müssen, wie sie in der Zeichenkette mit ihren Platzhaltern vorkommen. Werden die Platzhalter in der Zeichenkette geändert, müssen auch die Variablen am Ende neu sortiert werden.

In Swift wurde die Zusammensetzung von Zeichenketten neuentwickelt, so dass nicht mehr mit Platzhaltern, wie in Objective-C, gearbeitet werden muss. Es muss nur ein Backslash gefolgt von einer öffnenden und schließenden Klammer in der neuen Zeichenkette geschrieben werden. Innerhalb der Klammern wird die Variable geschrieben. Somit können mit weniger Aufwand Zeichenketten erstellt und auch gewartet werden.

Programmiersprache	Rang	
	September 2015	September 2016
Swift	16	13
Objective-C	10	14

Tabelle 3: Rangliste Programmiersprachen²⁶

Tabelle 3 zeigt die Rangliste vom September 2016 des TIOBE-Index. Der TIOBE-Index stellt die Häufigkeit dar, wie oft nach den Programmiersprachen gesucht wird. Die Auswahl der Suchmaschinen, die für die Auswertung beachtet werden, ist auf die 25 wichtigsten begrenzt.²⁷ Der Index beachtet zwar nur Anzahl der Suchaufträge für bestimmte Suchmaschinen, daran lässt sich aber dennoch ableiten, dass Swift mehr Interesse bekommt als Objective-C.

Für die Umsetzung des Projekts wurde sich bewusst für Swift entschieden. Swift erhält schon jetzt mehr Interesse als Objective-C. Langfristig werden dadurch mehr neue Projekte mit Swift entwickelt und Objective-C somit überholt. Die Bemühungen Swift für andere Plattformen zu öffnen, ist ein Schritt in die richtige Richtung. Inkompatibilität ist zwar noch ein Problem, der Quellcode lässt sich aber wahrscheinlich zu jeder neuen

²⁶ <http://www.tiobe.com/tiobe-index/> , Abbildung 1, eigene Darstellung

²⁷ Vgl.: <http://www.tiobe.com/tiobe-index/programming-languages-definition/> (16.09.2016)

Version relativ leicht beheben, da sich der Umfang des Quellcodes für den Prototyp der mobilen Anwendung im Rahmen befindet.

3.1.3 Serialisierung der Daten

Für die Datenübertragung von und zu einem Webserver gibt es verschiedene Datenformate. Dabei kann unterschieden werden, ob diese von Menschen lesbar sind oder nur von Maschinen.

Zwei der bekanntesten Formate sind *XML* und *JSON*. XML steht für „Extensible Markup Language“ und JSON steht für „JavaScript Object Notation“. Für die Entscheidung der Serialisierung wird sich auf XML und JSON beschränkt. Diese beiden Datenaustauschformate sind weit verbreitet. Durch die Bekanntheit beider Formate, ist die Unterstützung in vielen Programmiersprachen integriert. Daraus ergibt sich der Vorteil, dass keine Fremdsoftware genutzt werden muss, um die Funktionalität von JSON oder XML zu gewährleisten.

Um den Unterschied der beiden Datenformate zu verdeutlichen, folgen je ein Beispiel für XML und JSON.

```
<GetUserDataResponse>
  <GetUserDataResult>
    <FirstName>Max</FirstName>
    <LastName>Mustermann1036282</LastName>
    <PersonnelNumber>61191291</PersonnelNumber>
  </GetUserDataResult>
</GetUserDataResponse>
```

Abbildung 13: Beispiel XML Format

Abbildung 13 zeigt eine mögliche Antwort von einem Webserver. Die Antwort wurde in XML serialisiert. XML beginnt in diesem Beispiel mit dem Wurzelement „GetUserDataResponse“, gefolgt vom Elternelement „GetUserDataResult“. In dem Elternelement befinden sich die Textelemente, welche weiterverarbeitet werden. Die in spitze Klammern geschriebenen Worte werden Bezeichner genannt. Ein öffnender Bezeichner ist in diesem Beispiel „<FirstName>“ und der dazugehörige schließende Bezeichner ist „</FirstName>“. Dazwischen befindet sich das Textelement, welches zur Weiterverarbeitung benötigt wird. Die Zeichenanzahl für dieses Beispiel beträgt 189.

```
{
  "GetUserDataResult": {
    "FirstName": "Max",
    "LastName": "Mustermann1036282",
    "PersonnelNumber": "61191291"
  }
}
```

Abbildung 14: Beispiel JSON Format

In Abbildung 14 wird ein JSON Objekt dargestellt. Das Objekt beginnt mit einer geschweiften Klammer. Danach folgt das Objekt „GetUserDataResult“, welches Eigenschaften enthält. Die Eigenschaften des Objekts bestehen aus einem Schlüssel und einem zugewiesenen Wert. „FirstName“ ist ein Schlüssel mit dem Wert „Max“. Die Zeichenanzahl dieses Beispiels beträgt 101.

Durch den Vergleich beider Beispiele wird erkennbar, dass für die Übermittlung derselben Informationen, mit JSON weniger Zeichen übertragen werden müssen. Mit XML werden 189 Zeichen übertragen, was 189 Bytes entspricht. Im Gegensatz zu JSON, wo nur 101 Bytes übertragen werden müssen, ergibt sich eine Differenz von 88 Bytes. Der Grund für die erhöhte Zeichenanzahl sind die Bezeichner, welche zum Öffnen und zum Schließen vorkommen. In JSON gibt es keine schließenden Bezeichner.

Werden beide Werte in Relation gesetzt, ergibt sich eine Einsparung von 47 Prozent, wenn JSON statt XML eingesetzt wird. Die Werte unterscheiden sich zwar je nach Länge und Formatierung der Datenobjekte, in der Regel reduzieren sich aber die Daten für die Übertragung bei der Verwendung von JSON.

Übertragungstechnik	Übertragungs- geschwindigkeit	Übertragungsdauer von		
		88 Byte	300 Kilobyte	375 Kilobyte
		in Sekunden		
GPRS (General Packet Radio Service)	54 kBit/s	< 1	44	56
EDGE (Enhanced Data Rates for GSM Evolution)	236 kBit/s	< 1	10	13
UMTS (Universal Mobile Telecommunications System)	384 kBit/s	< 1	6	8
HSDPA (High Speed Downlink Packet Access)	7,2 Mbit/s	< 1	< 1	< 1

Tabelle 4: Vergleich Mobilfunkstandards

Tabelle 4 veranschaulicht die Unterschiede verschiedener Übertragungstechniken hinsichtlich der Übertragungsdauer für eine bestimmte Menge an Daten. In der ersten Spalte werden die Übertragungstechniken genannt. In Spalte zwei sind die zugehörigen theoretischen Übertragungsgeschwindigkeiten für den Download aufgeführt. Download bedeutet, Daten werden von einem Webserver zu dem Gerät übertragen. Die dritte Spalte gibt die Übertragungsdauer an. Sie ist dabei in drei Teile untergliedert. Es wird jeweils die Dauer für Übertragung von 88 Byte, 300 Kilobyte und 375 Kilobyte in Sekunden angegeben.

Für das erste Beispiel wurde die Zeichendifferenz von Abbildung 13 und Abbildung 14 gewählt. Es wird ersichtlich, dass es bei sehr wenigen Zeichen keine Rolle spielt, ob die Daten mit JSON oder XML serialisiert wurden, da es auch mit einer langsamen Übertragung weniger als eine Sekunde dauert. Allerdings wird der Unterschied wahrnehmbar, wenn die zu übertragende Datenmenge größer wird. Deshalb wurden die Werte 300 Kilobyte und 375 Kilobyte gewählt. Es wird angenommen, dass eine Beispielübertragung mit JSON 300 Kilobyte beträgt. Sind die Daten mit XML serialisiert, liegt die zu übertragende Menge häufig über dem JSON Wert. Deshalb wurden für das Beispiel 25 Prozent Überschuss gewählt, was 375 Kilobyte ergibt. Mit schnellen Übertragungstechniken, wie HSDPA, sind kaum Unterschiede spürbar. Je langsamer die Datenrate wird, umso größer ist der Unterschied. Steht nur GPRS zur Verfügung, beträgt die Zeitdifferenz ca. zwölf Sekunden. Daran lässt sich erkennen, dass die Seriali-

sierung der Daten von wichtiger Bedeutung ist. Durch Einsparung der Datenmenge werden Datenverkehr und Übertragungsdauer reduziert.

3.2 Anforderungen

Das vorhandene Bildungsmanagementsystem ist für die Kommunikation mit mobilen Endgeräten nicht geeignet. Die gegenwärtigen Methoden zur Übertragung der Daten vom Webserver zum Client enthalten Informationen, welche für den Prototyp nicht benötigt werden. Dieser Ballast verursacht unnötigen Datenverkehr. Es werden neue Methoden entwickelt, welche nur die nötigsten Informationen übermitteln, um eine schnelle Kommunikation zwischen Webserver und Client zu ermöglichen.

Die Anforderungen an die Funktionen des Webserver werden in der nachfolgenden Tabelle aufgeführt.

Anforderung	Bezeichnung	Beschreibung
1.	Login	<ul style="list-style-type: none">▪ Überprüfung der entgegengenommenen Anmeldeinformationen▪ Rückgabe, ob Anmeldeinformationen richtig oder falsch
2.	Benutzerdaten	<ul style="list-style-type: none">▪ Übertragung der Benutzerdaten des angemeldeten Anwenders
3.	Registrierungen	<ul style="list-style-type: none">▪ Rückgabe der Kurse, in welche der Anwender eingeschrieben ist▪ beendete Kurse werden nicht angezeigt
4.	Gesamtangebot	<ul style="list-style-type: none">▪ Durchsuchen des Gesamtangebots mit bestimmten Parametern:<ul style="list-style-type: none">○ Suchbegriff (Optional)○ Angebotstyp▪ Rückgabe der verfügbaren Kurse, welche auf die Kriterien zutreffen

Tabelle 5: Anforderungen an Webserver

Tabelle 5 beschreibt die Anforderungen an den Webserver. Insgesamt gibt es vier Funktionen, welche der Server haben muss. Für jede Funktion gibt die Tabelle die Bezeichnung und die dazugehörige Beschreibung wieder.

Die Anforderungen an die mobile Anwendung gliedern sich in Anforderungen an die Funktionalität und Benutzeroberfläche.

Anforderung	Bezeichnung	Beschreibung der Oberfläche
1.	Login	<ul style="list-style-type: none">▪ zwei Textfelder für Benutzername und Passwort▪ ein Button zum Anmelden
2.	Benutzerdaten	<ul style="list-style-type: none">▪ übersichtliche Ausgabe der Benutzerdaten
3.	Registrierungen	<ul style="list-style-type: none">▪ Kurse, in welche der Benutzer eingeschrieben ist, werden in einer Tabelle ausgegeben
4.	Registrierungs- details	<ul style="list-style-type: none">▪ Ausgabe genauer Informationen zu einer Registrierung
5.	Gesamtangebot durchsuchen	<ul style="list-style-type: none">▪ ein Textfeld, in welches der Suchbegriff eingetragen werden kann▪ Möglichkeit zur Spezifizierung des Angebots
6.	Suchergebnis	<ul style="list-style-type: none">▪ Ausgabe aller auf die Sucheinstellung zutreffenden Kurse in einer Tabelle
7.	Suchergebnis- details	<ul style="list-style-type: none">▪ Ausgabe zusätzlicher Informationen

Tabelle 6: Anforderungen an Oberfläche des Prototyps

In Tabelle 6 werden die Anforderungen an die Benutzeroberfläche dargestellt. Dabei wird nur festgelegt, was auf der Oberfläche zu sehen sein soll. Für den Prototyp gibt es keine festen Bestimmungen zur Anordnung der Elemente.

Anforderung	Bezeichnung	Beschreibung der Funktionen
1.	Login	<ul style="list-style-type: none"> ▪ Betätigung des Buttons prüft, ob Anmelde-daten korrekt sind ▪ Benutzer erhält eine Meldung, falls Anmel-daten inkorrekt sind ▪ leitet bei erfolgreicher Anmeldung zur nächsten Oberfläche weiter
2.	Benutzerdaten	<ul style="list-style-type: none"> ▪ zeigt nur die Benutzerdaten an, keine weitere Interaktion möglich
3.	Registrierungen	<ul style="list-style-type: none"> ▪ die Auswahl eines Tabelleneintrags führt den Benutzer zu den Registrierungsdetails
4.	Registrierungs-details	<ul style="list-style-type: none"> ▪ Benutzer kann zurück zur Registrierungs-übersicht wechseln
5.	Gesamtangebot durchsuchen	<ul style="list-style-type: none"> ▪ durchsucht den Kurskatalog nach denen vom Benutzer angegeben Parametern
6.	Suchergebnis	<ul style="list-style-type: none"> ▪ Ausgabe aller auf die Sucheinstellung zu-treffenden Kurse in einer Tabelle
7.	Suchergeb-nisdetails	<ul style="list-style-type: none"> ▪ Anwender kann zu den Suchergebnissen zurückwechseln

Tabelle 7: Anforderungen an Funktionen des Prototyps

Die Anforderungen an die Funktionen des Prototyps werden in Tabelle 7 aufgeführt. Jede Funktion hat eine Bezeichnung und eine Beschreibung, was sie beherrschen muss.

3.3 Realisierung des Prototyps

Die praktische Umsetzung des Prototyps ist in zwei Teile gegliedert. Zuerst wird die Programmierung der Software auf dem Webserver realisiert. Nach erfolgreicher Umsetzung des Backends, wird die mobile Anwendung entwickelt.

3.3.1 Kommunikationsweg

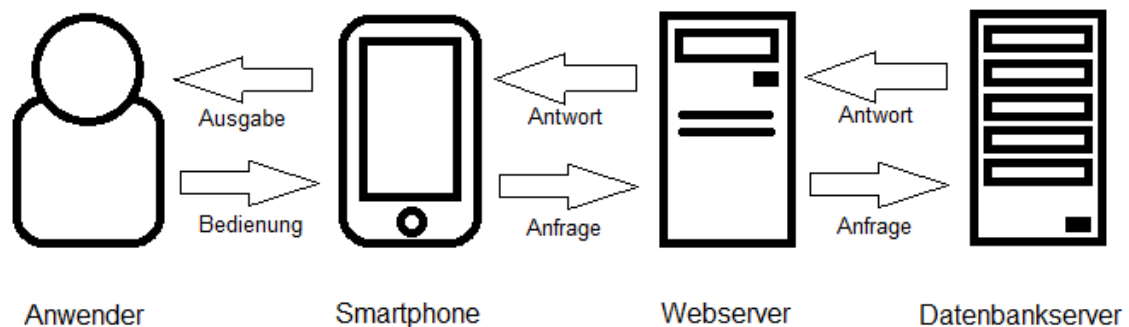


Abbildung 15: Kommunikationsweg

Die Grafik zeigt den Kommunikationsweg für die Anwendung. Der Anwender bedient das Smartphone. Je nach Funktion, welche vom Benutzer ausgewählt wird, sendet das Smartphone eine Anfrage zum Webserver. Die Software auf dem Webserver verarbeitet die Anfrage. Der Webserver ist mit dem Datenbankserver verknüpft und stellt eine Anfrage an diesen. Aus der Datenbank werden die nötigen Informationen an den Webserver übermittelt. Anschließend arbeitet der Webserver die Daten aus der Datenbank auf und übergibt sie als Antwort an das mobile Endgerät. Das Gerät stellt die Informationen auf dem Display dar, welche vom Benutzer gelesen werden können.

3.3.2 Methoden des Webserver

Wie in „Tabelle 5: Anforderungen an Webserver“ in dem Absatz „Anforderungen“ aufgezeigt, bestehen vier Bedingungen an die Funktionalität des Webserver. Die vier Anforderungen werden in jeweils einer Methode umgesetzt.

Der Webserver ist auf dem Testsystem erreichbar unter „<http://127.0.0.1/BmsMobileAPI/MobileAPIService.svc>“. Je nach Methode, welche aufgerufen werden soll, wird an die URL eine bestimmte Bezeichnung angefügt.

URL	Parameter	Beschreibung der Funktion
/Login	<ul style="list-style-type: none">▪ Benutzername▪ Passwort	<ul style="list-style-type: none">▪ Prüft, ob die Anmeldedaten korrekt sind und gibt wahr oder falsch zurück
/GetUserData	<ul style="list-style-type: none">▪ Benutzername▪ Passwort	<ul style="list-style-type: none">▪ Gibt für den jeweiligen Nutzer die Benutzerdaten zurück
/GetRegistrations	<ul style="list-style-type: none">▪ Benutzername▪ Passwort	<ul style="list-style-type: none">▪ Gibt alle Kurse zurück, für welche der Benutzer registriert ist
/GetCourses	<ul style="list-style-type: none">▪ Benutzername▪ Passwort▪ Suchbegriff (Optional)▪ Angebotstyp	<ul style="list-style-type: none">▪ Gibt alle Kurse zurück, welche mit dem Suchbegriff und dem Angebotstyp übereinstimmen

Tabelle 8: Funktionen des Webservers

Tabelle 8 beschreibt die Funktionen des Webservers. Die erste Spalte gibt Auskunft, über welche URL die Funktion zu erreichen ist. In der zweiten Spalte sind die Parameter zu finden, welche bei dem Aufruf der Funktion übergeben werden müssen, um die Funktion ausführen zu können. Die letzte Spalte beschreibt, was die Funktion ausführt.

3.3.3 JSON Objekte

Die Methoden des Webservers in Tabelle 8 nehmen jeweils ein JSON Objekt entgegen, verarbeiten es und liefern ein JSON Objekt mit den ermittelten Daten zurück.

JSON Objekte der Anfragen an den Webserver

Methode	JSON Objekt und Beschreibung
Login	<pre>{ "username" : "BENUTZERNAME", "password" : "PASSWORT" }</pre> <ul style="list-style-type: none"> ▪ Objekt besteht aus Benutzername und Passwort ▪ wird bei dem Login vom Benutzer eingegeben ▪ Anmeldedaten werden nach erfolgreicher Anmeldung auf dem Gerät gespeichert
GetUserData	<pre>{ "username" : "BENUTZERNAME", "password" : "PASSWORT" }</pre> <ul style="list-style-type: none"> ▪ Objekt besteht aus Benutzername und Passwort ▪ wird nicht vom Anwender eingegeben ▪ Objekt wird aus den gespeicherten Daten generiert
GetRegistrationData	<pre>{ "username" : "BENUTZERNAME", "password" : "PASSWORT" }</pre> <ul style="list-style-type: none"> ▪ Objekt besteht aus Benutzername und Passwort ▪ wird nicht vom Anwender eingegeben ▪ Objekt wird aus den gespeicherten Daten generiert
GetCourses	<pre>{ "username" : "BENUTZERNAME", "password" : "PASSWORT", "query" : "SUCHBEGRIFF", "offerType" : ANGEBOTSTYP }</pre> <ul style="list-style-type: none"> ▪ Objekt besteht aus Benutzername, Passwort, Suchbegriff und Angebotstyp ▪ Benutzername und Passwort werden aus den gespeicherten Daten übernommen ▪ Suchbegriff ist optional und kann vom Anwender eingegeben werden ▪ Angebotstyp wird vom Anwender über eine Auswahl auf der Oberfläche getroffen

Tabelle 9: Erläuterung der JSON Objekte der Anfragen

Die JSON Objekte, welche als Anfragen vom Gerät zu dem Webserver geschickt werden, sind in Tabelle 9 dargestellt. Für jede Methode ist das mit JSON serialisierte Objekt beginnend mit einer öffnenden geschweiften Klammer und endend mit einer schließenden geschweiften Klammer dargestellt. In Großbuchstaben geschriebene Wörter sind Variablen, welche vom Gerät entsprechend gesetzt werden. Unter dem JSON Objekt befindet sich eine Erläuterung.

JSON Objekte der Antworten vom Webserver

Die folgenden Werte der JSON Objekte sind fiktive Daten.

Methode	JSON Objekt und Beschreibung
Login	<pre>{ "IsLoginCorrectResult": true }</pre> <ul style="list-style-type: none">bei erfolgreicher Anmeldung, liefert der Webserver „true“, sonst „false“
GetUserData	<pre>{ "GetUserDataResult": { "DateOfBirth": "23.07.1973", "Email": "max.mustermann1036282@local.local", "FirstName": "Max", "JobName": "PHK", "LastName": "Mustermann1036282", "OrgUnitName": "BWKA-DR-PR-KAWALDST-DGR", "PersonnelNumber": "61191291", "PhoneNumber": "0123 56789-0", "Salutation": "Herr", "Title": "" } }</pre> <ul style="list-style-type: none">das Objekt besteht aus:<ul style="list-style-type: none">GeburtsdatumE-Mail-AdresseVornameDienst-/AmtsbezeichnungNachnameDienststellePersonalnummerTelefonnummerAnredeTitel

Methode	JSON Objekt und Beschreibung
GetRegistrationData	<pre>{ "GetRegistrationsResult": [{ "BriefDescription": "AZT EKA-2", "CourseBegin": "13.10.2016 08:30", "CourseEnd": "13.10.2016 10:30", "CourseLocation": "KA ET", "Title": "EKA Training" }] }</pre> <ul style="list-style-type: none"> ▪ das Objekt besteht aus: <ul style="list-style-type: none"> ○ Kurzform des Titels ○ Kursbeginn ○ Kursende ○ Kurstitel ▪ das Objekt kann aus mehreren Einträgen bestehen
GetCourses	<pre>{ "GetCoursesResult": [{ "Begin": "10.10.2016 10:00", "End": "14.10.2016 12:25", "OfferTypeld": 1, "ParticipantsCurrent": 20, "ParticipantsMax": 20, "Title": "Kriminaltechnik - Modul Kriminaltechnische Brandursachenermittlung" }] }</pre> <ul style="list-style-type: none"> ▪ das Objekt besteht aus: <ul style="list-style-type: none"> ○ Kursbeginn ○ Kursende ○ Angebotstyp ○ Anzahl der eingeschriebenen Teilnehmer ○ Anzahl der maximalen Teilnehmer ○ Kurstitel ▪ das Objekt kann aus mehreren Einträgen bestehen

Tabelle 10: Erläuterung der JSON Objekte der Antworten

Tabelle 10 ist genauso aufgebaut wie Tabelle 9. Hier werden allerdings die Antworten, welche vom Webserver ermittelt und an das Gerät gesendet werden aufgezeigt. Um die Übersicht zu wahren, wurden bei den Methoden „GetRegistrationData“ und „GetCourses“ nur jeweils eine Registrierung bzw. ein Kurs dargestellt. Je nach Anwendungsfall, können diese JSON Objekte mit mehreren Einträgen gefüllt sein.

3.3.4 Oberfläche der mobilen Anwendung

Die App besteht aus mehreren Views. Als Views werden die Oberflächen bezeichnet, welche auf dem Smartphone dargestellt werden.

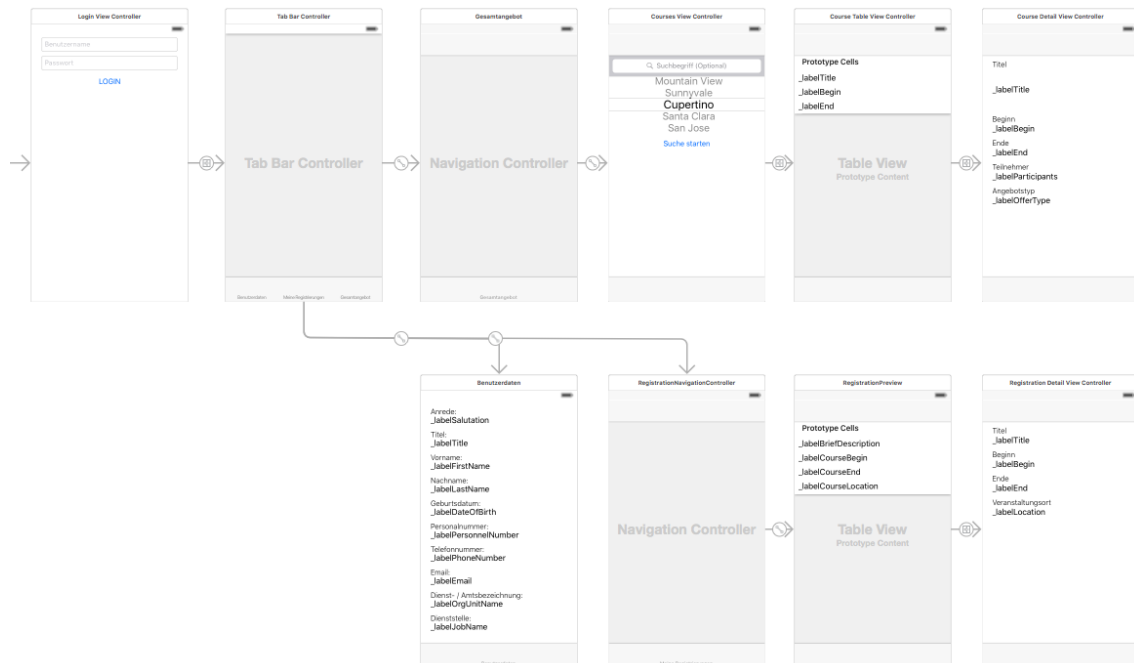


Abbildung 16: Überblick aller Views

Abbildung 16 stellt alle Views der Anwendung dar. Die App ist in vier Hauptbereiche geteilt. Bei dem Start der App erscheint die Anmeldemaske. Bei erfolgreicher Anmeldung, hat der Anwender Zugriff auf die drei weiteren Bereiche. Diese sind unterteilt in die Anzeige der Benutzerdaten, der Registrierungen und des Gesamtangebots.

Ansicht der Anmeldung

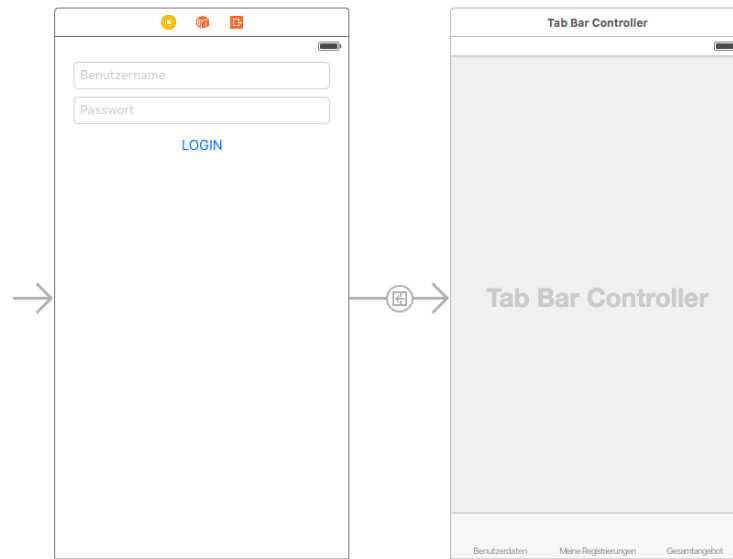


Abbildung 17: Ansicht der Anmeldung

Öffnet der Anwender die Applikation auf seinem Mobilgerät, erscheint eine Ansicht wie in Abbildung 17. Der Benutzer muss seine Anmeldedaten eingeben und wird bei erfolgreicher Anmeldung zu der nächsten Ansicht weitergeleitet. Die folgenden Ansichten sind in einem „Tab Bar Controller“ gebettet. Das bedeutet, dass in dem unteren Bereich des Displays eine Leiste ist, von der aus in die anderen Ansichten gewechselt werden kann.

Fehlermeldungen bei der Anmeldung

Sollten bei der Anmeldung Fehler auftreten, wird dem Benutzer, je nach Fehler, eine entsprechende Meldung ausgegeben.

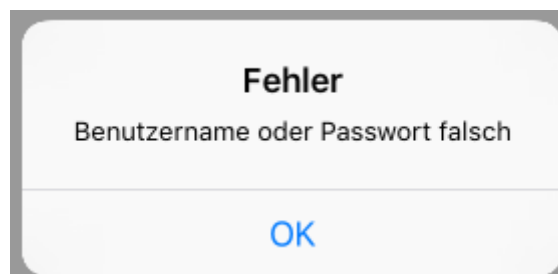


Abbildung 18: Fehlermeldung Anmeldung I

Diese Fehlermeldung erscheint, wenn der Benutzer falsche Anmeldedaten eingegeben hat.

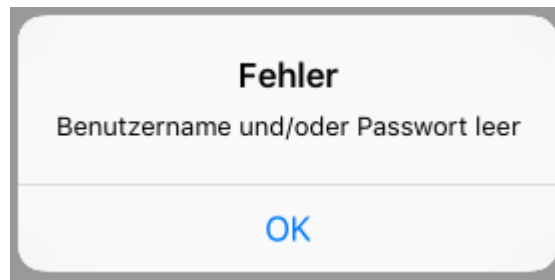


Abbildung 19: Fehlermeldung Anmeldung II

Hat der Anwender bei dem Anmeldeversuch ein oder beide Felder frei gelassen, wird dieser Fehler ausgegeben.

Ansicht der Benutzerdaten

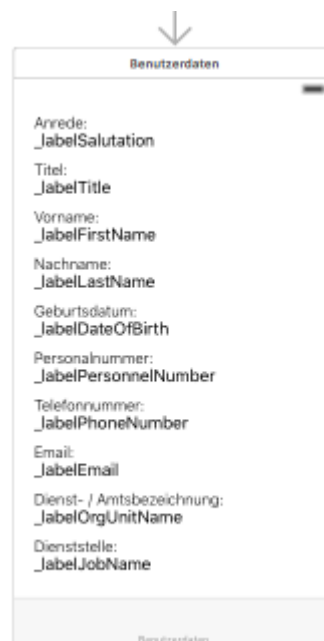


Abbildung 20: Ansicht der Benutzerdaten

Auf der Ansicht der Benutzerdaten sind keine Interaktionen möglich. Der Anwender sieht nur die Benutzerdaten, welche in seinem Benutzerkonto hinterlegt sind.

Ansicht der Registrierungen

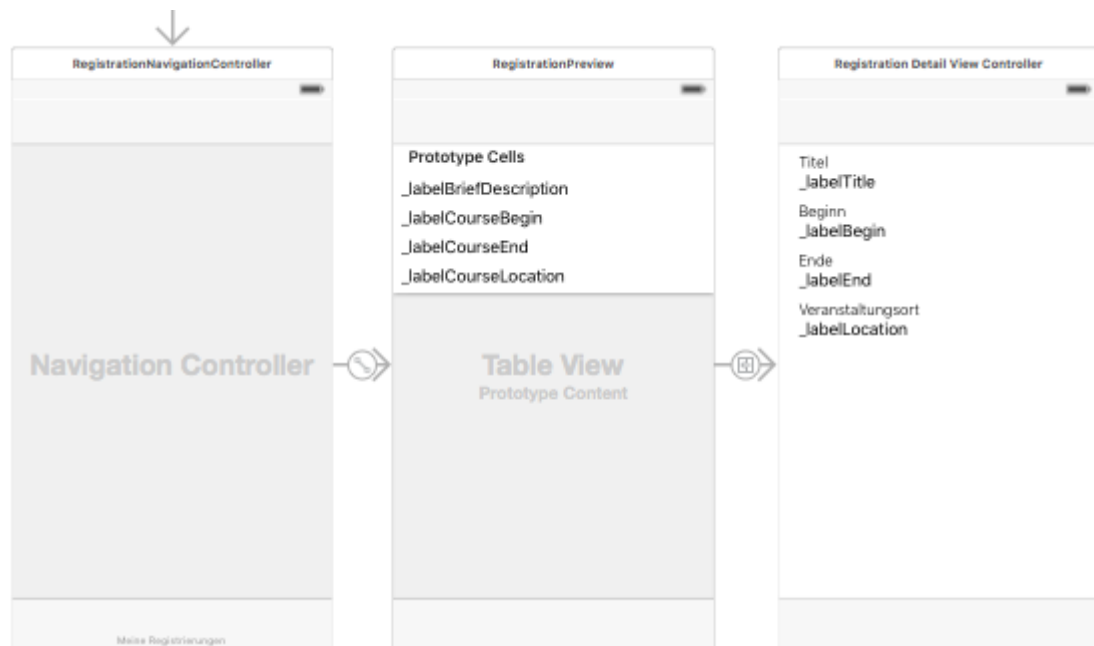


Abbildung 21: Ansicht der Registrierungen

Diese Oberfläche besteht aus mehreren Ansichten. Wird diese View vom Benutzer angewählt, werden alle Kurse ausgelesen, für welche der Benutzer registriert ist. Die Ergebnisse werden in Form einer Tabelle angezeigt. Die Tabelle besteht nur aus einer Spalte. Das Aussehen der Tabellenzellen wurde mit sogenannten „Prototype Cells“ festgelegt, was in Abbildung 21 in der zweiten Ansicht abgebildet ist. Der Inhalt einer Zelle besteht aus der Kurzbeschreibung des Kurses, Kursbeginn, Kursende und Veranstaltungsort.

Durch Berühren einer Zelle, wird die Detailansicht zu dem ausgewählten Kurs angezeigt. Die Detailansicht besteht aus dem vollständigen Titel des Kurses, Kursbeginn, Kursende und Veranstaltungsort. Der Anwender hat die Möglichkeit, wieder zurück zur Tabelle zu gelangen, um sich seine weiteren Registrierungen anzuschauen.

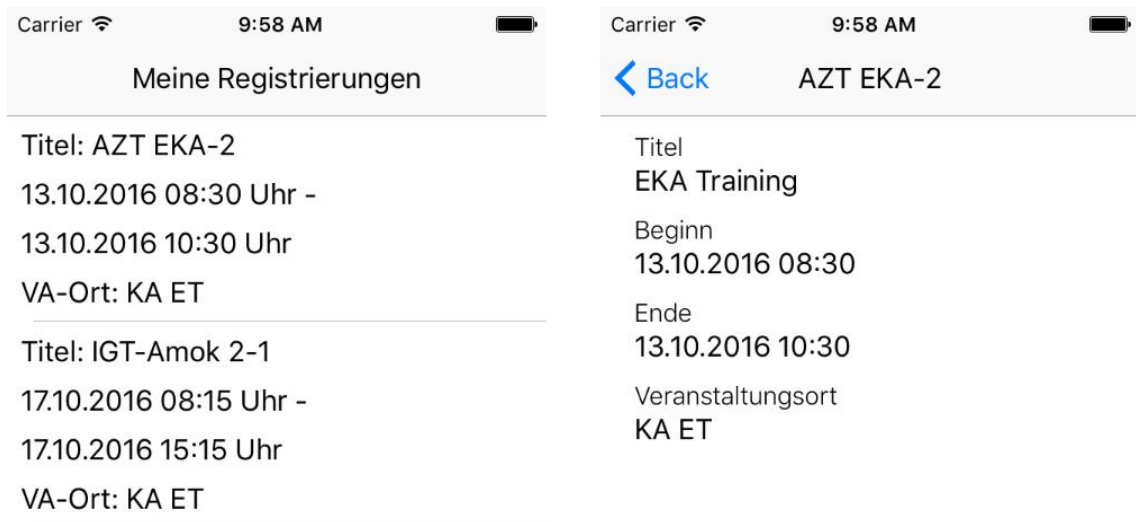


Abbildung 22: Ansicht der Registrierungen im Simulator

Um die Ansicht der Ausgabe der Registrierungen anschaulicher zu gestalten, wurde Abbildung 22 eingefügt. Die Grafik zeigt zwei Ausschnitte von der Ausführung im Simulator. Auf der linken Seite ist die Liste mit den Registrierungen dargestellt. Bei der Berührung eines Eintrags, wird die Detailansicht geöffnet, welche auf der rechten Seite dargestellt wird. Oben links mit „< Back“ kann der Anwender zurück in die Registrierungen wechseln.

Ansicht des Gesamtangebots

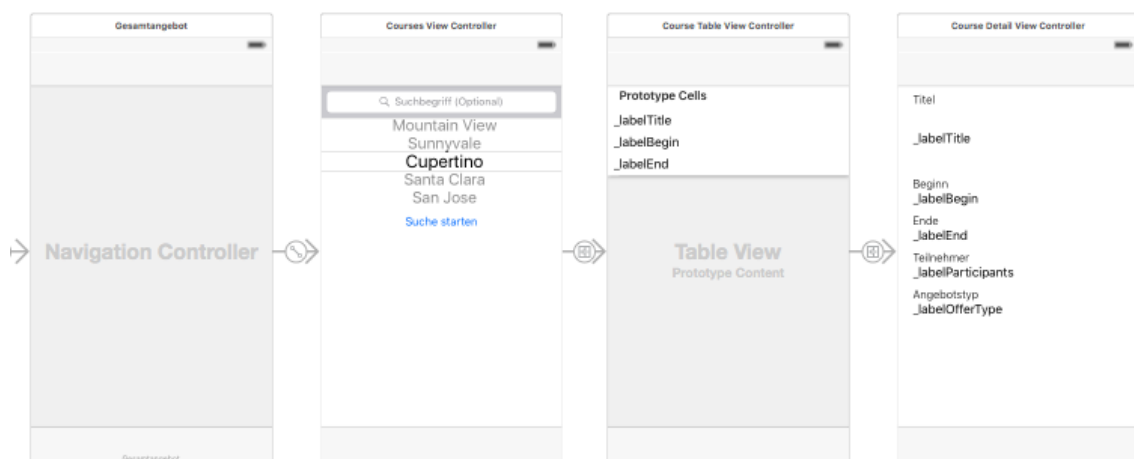


Abbildung 23: Ansicht des Gesamtangebots

Der Menüpunkt „Gesamtangebot“ ist ähnlich aufgebaut, wie die Ansicht der Registrierungen. In diesem Menü kann das Kursangebot durchsucht werden. Der Anwender hat die Möglichkeit, einen Suchbegriff einzugeben sowie den Angebotstyp auszuwählen, bevor die Suche gestartet wird. Sobald der Button „Suche starten“ angetippt wird, sendet die App eine Anfrage an den Webserver. Der Webserver gibt im Anschluss eine Antwort an die App mit den Kursen, welche auf die Suchanfrage zutreffen. Diese werden in einer Tabelle aufgelistet. Wie bei den Registrierungen, werden die Tabellenzellen in der Übersicht mit wenigen Daten befüllt, damit die Tabelle übersichtlicher bleibt. Die Zellen bestehen aus dem Kurstitel, Kursbeginn und Kursende. Wählt der Benutzer eine Zelle aus, werden Details zu dem Kurs angezeigt. Die Detailansicht besteht aus dem Kurstitel, Kursbeginn, Kursende, aktuelle Teilnehmeranzahl, maximale Teilnehmeranzahl und Angebotstyp.

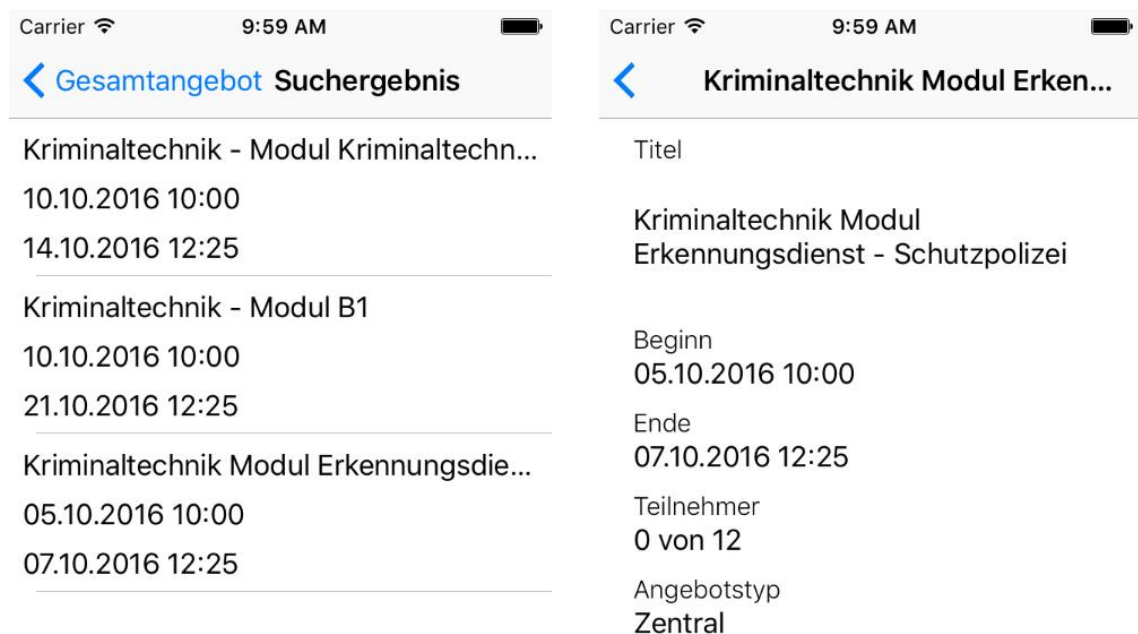


Abbildung 24: Ansicht der Suchergebnisse im Simulator

Nach dem gleichen Prinzip wie die Ausgabe der Registrierungen, ist die Ausgabe der Suchergebnisse aufgebaut. Abbildung 24 zeigt auf der linken Seite Suchergebnisse in einer Tabelle an. Auf der rechten Seite sind die Details für das ausgewählte Suchergebnis.

4 Zusammenfassung und Weiterentwicklung

4.1 Zusammenfassung

Das Ziel dieser Arbeit konnte erreicht werden. Es wurde eine lauffähige Software entwickelt. Der Prototyp besitzt die Möglichkeit, sich mit dem Webserver zu verbinden, um Daten auf dem mobilen Endgerät anzeigen zu lassen. Benutzerdefinierte Anfragen werden auf dem Server verarbeitet und eine Antwort an das Gerät verschickt.

Außerdem sollte evaluiert werden, inwiefern sich ein Bildungsmanagementsystem in der Cloud nutzen lässt. Bildungsmanagement in die Cloud auszulagern kann für private Schulen und kleine Unternehmen eine sinnvolle Alternative darstellen. Der Ressourcenaufwand wird dadurch vermindert, da der Cloud-Anbieter für alle Aufgaben zum Betrieb der Rechenzentren verantwortlich ist.

4.2 Weiterentwicklung

Die Software des Webserver sowie die mobile Anwendung bieten viele Möglichkeiten für Weiterentwicklungen. Durch Verwendung der objektorientierten Softwareentwicklung, ist der Quellcode so strukturiert, dass weitere Anpassungen ohne große Änderungen an der Struktur möglich sind.

Weiterentwicklung des Webserver

Die *API* wurde zwar für die mobile Anwendung entworfen. Der Webserver hat aber vorgefertigte Methoden, um auf die Datenbank zuzugreifen. Diese Abfragen überprüfen im Hintergrund viele Entscheidungskriterien, die nicht alle für die Antwort benötigt werden. Es kommt dadurch zu längeren Rechenzeiten und mehr Auslastung.

Alle Anfragen an den Webserver müssen mit Benutzername und Passwort gesendet werden. Für solche Anmeldefunktionen gibt es Lösungen, in welchen Authentifizierungstoken eingesetzt werden. Dafür muss eine Datenbank existieren, worin der Token gespeichert wird. In fremden drahtlosen Netzwerken können dritte die Kommunikation abhören. Wird über eine unsichere HTTP Verbindung für die Kommunikation zum Webserver genutzt, können unter Umständen die Benutzerdaten ausgelesen werden. Ein Token erschwert dem Angreifer den Zugriff.

Die Übertragung von Daten an das Gerät ist ein weiterer Punkt, welcher verbessert werden kann. Zum Beispiel bei der Suche nach verfügbaren Kursen. Der Webserver ermittelt in der Datenbank, wie viele Kurse auf die Suchanfrage zutreffen. Im Anschluss werden alle Kurse an das Gerät transportiert. In diesem Schritt ist eine sinnvolle Lösung, die Kurse nach und nach zu übertragen, um die Wartezeit für den Anwender und auch die Rechenleistung zu minimieren.

Mit dem Prototyp ist es auch noch nicht möglich, Kurse zu buchen. In zukünftigen Versionen sollte über eine Implementierung dieser Funktion nachgedacht werden. Zurzeit ist es nur möglich sich gebuchte Kurse und verfügbare Kurse anzuschauen.

Weiterentwicklung der mobilen Anwendung

Die meisten Punkte der Weiterentwicklung betreffen den Webserver. Finden Änderungen an der API statt, können diese einfach in die Anwendung eingepflegt werden.

Weiterentwicklungen an der App müssen zuerst vom Webserver unterstützt werden, bevor diese in der Applikation realisiert werden können. Das liegt daran, dass der Server die veränderten JSON Objekte nicht als Anfrage akzeptiert. Es ist nicht möglich, weitere Filtermöglichkeiten in der App anzubieten, bevor der Webserver diese auch entgegennehmen kann.

Wurden die Ideen zur Weiterentwicklung umgesetzt, muss überprüft werden, ob die für den Prototyp gewählte Gestaltung der Benutzeroberfläche, noch geeignet ist. Durch Hinzufügen neuer Funktionen kann eine Inkompatibilität mit dem aktuellen Design entstehen.

Glossar

Begriff	Erklärung
Android	von Google entwickeltes mobiles Betriebssystem
API	Schnittstelle zur Anwendungsprogrammierung
Git	Software zur Versionsverwaltung
GitHub	Online-Dienst, welcher kollaborative Versionsverwaltung mit <i>Git</i> ermöglicht
iOS	von Apple entwickeltes mobiles Betriebssystem.
Framework	bezeichnet in der Softwareentwicklung ein Programmiergerüst
JSON	JavaScript Object Orientation, Verfahren zur Serialisierung von Daten
XML	Extensible Markup Language, Auszeichnungssprache, unter anderem zur Serialisierung von Daten
View	bezeichnet eine Oberfläche bei der iOS Programmierung

Literaturverzeichnis

o.V. <keine Email> : Bildungsmanagement - Prozess und Aufgaben. URL: <<http://www.diebildungsmanager.at/blog/bildungsmanagement/>>, verfügbar am 15.09.2016

o.V. <keine Email> : ohne Namen. URL:<<http://www.apfelwiki.de/briefkasten/Main/nokia9300i.jpg>>, verfügbar am 06.08.2016

Michael Arrington <keine Email> : iPhone App Store Has Launched (Updated). URL: <<https://techcrunch.com/2008/07/10/app-store-launches-upgrade-itunes-now/>>, verfügbar am 06.08.2016

o.V. <keine Email> : Apple Reinvents the Phone with iPhone. URL: <<http://www.apple.com/pr/library/2007/01/09Apple-Reinvents-the-Phone-with-iPhone.html>>, verfügbar am 06.08.2016

o.V. <keine Email> : ohne Namen. URL: <<https://www.iphoneshopusa.com/c/12-category/iphone-2g.jpg>>, verfügbar am 06.08.2016

Statista <support@statista.com> : Number of available apps in the Apple App Store from July 2008 to June 2016. URL: <<https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/>>, verfügbar am 06.08.2016

Daniel Kuhn <keine Email> : HTC Dream: Erstes Android-Smartphone wurde vor 5 Jahren vorgestellt. URL: <<http://www.giga.de/smartphones/htc-dream/news/htc-dream-erstes-android-smartphone-wurde-vor-5-jahren-vorgestellt/>>, verfügbar am 06.08.2016

Melissa Perenson <keine Email> : Google Launches Android Market. URL: <http://www.pcworld.com/article/152613/google_android_ships.html>, verfügbar am 06.08.2016

Chris Velazco <keine Email> : Goodbye Android Market, Hello Google Play. URL: <<https://techcrunch.com/2012/03/06/goodbye-android-market-hello-google-play/>>, verfügbar am 06.08.2016

Statista <support@statista.com> : Number of available applications in the Google Play Store from December 2009 to September 2016. URL: <<https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>>, verfügbar am 06.08.2016

o.V. <keine Email> : kein Namen. URL: <<https://www.draw.io/>>, verfügbar am 06.08.2016

TB <keine Email> : Cloud-Computing – Oder was ist eigentlich die Cloud?. URL: <<https://blog.btfrei.de/2012/12/cloud-computing-oder-was-ist-eigentlich-die-cloud/>>, verfügbar am 06.08.2016

Jens Ihlenfeld <keine Email> : Amazon erklärt Ausfall seiner Cloud-Server. URL: <<http://www.golem.de/1104/83140.html>>, verfügbar am 07.08.2016

Bitkom <a.pols@bitkom-research.de> : Cloud-Monitor 2015. URL: <<https://www.bitkom.org/Publikationen/2015/Studien/Cloud-Monitor-2015/Cloud-Monitor-2015-KPMG-Bitkom-Research.pdf>>, verfügbar am 07.08.2016

o.V. <keine Email> : Microsoft Cloud Deutschland. URL: <<https://www.microsoft-politik.de/de-de/politik/themen/cloud-deutschland.aspx>>, verfügbar am 07.08.2016

LEONARD KLINT <keine Email> : Microsoft Azure: Die „Deutschland Cloud“ ist da. URL: <<https://windowsunited.de/2016/09/21/microsoft-azure-die-deutschland-cloud-ist-da/>>, verfügbar am 27.09.2016

Netmarketshare <keine Email> : Mobile/Tablet Operating System Market Share. URL: <<https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1>>, verfügbar am 15.09.2016

o.V. <keine Email> : Dashboards. URL: <<https://developer.android.com/about/dashboards/index.html>>, verfügbar am 15.09.2016

o.V. <keine Email> : Android 7.0 Nougat: a more powerful OS, made for you. URL: <<https://android.googleblog.com/2016/08/android-70-nougat-more-powerful-os-made.html>>, verfügbar am 15.09.2016

Will Verduzco <keine Email> : Here's Everything That's Changed in Android 4.4.4 KTU84P. URL: <<http://www.xda-developers.com/heres-everything-thats-changed-in-android-4-4-4-ktu84p/>>, verfügbar am 15.09.2016

Chris Chavez <keine Email> : Android 5.1.1 (LMY47V) officially uploaded to AOSP and here's the full changelog, Nexus Player factory images also available. URL: <<http://phandroid.com/2015/04/21/android-5-1-1-lmy47v-aosp-changelog-nexus-player-factory-images/>>, verfügbar am 15.09.2016

changelog-nexus-player-factory-images/>, verfügbar am 15.09.2016

o.V. <keine Email> : App Store. URL: <<https://developer.apple.com/support/app-store/>>, verfügbar am 15.09.2016

o.V. <keine Email> : keinen Namen. URL: <<http://www.apple.com/de/swift/>>, verfügbar am 15.09.2016

o.V. <keine Email> : Swift is Open Source. URL: <<https://developer.apple.com/swift/blog/?id=34>>, verfügbar am 15.09.2016

Ted Kremenek <kremenek@apple.com> : Swift 3.0 Released!. URL: <<https://swift.org/blog/swift-3-0-released/>>, verfügbar am 15.09.2016

TIOBE <info@tiobe.com> : TIOBE Index for August 2016. URL: <<http://www.tiobe.com/tiobe-index/>>, verfügbar am 16.09.2016

TIOBE <info@tiobe.com> : TIOBE Programming Community Index Definition. URL: <<http://www.tiobe.com/tiobe-index/programming-languages-definition/>>, verfügbar am 16.09.2016

Sascha Kersken <keine Email> : Kompendium der Informationstechnik. URL:
<<http://openbook.rheinwerk-verlag.de/kit/itkomp15000.htm>>, verfügbar am 19.09.2016

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Vorname Nachname